


[Subscribe \(Full Service\)](#) [Register \(Limited Service, Free\)](#) [Login](#)

 Search: ☒ The ACM Digital Library ☐ The Guide

THE ACM DIGITAL LIBRARY


[Feedback](#) [Report a problem](#) [Satisfaction survey](#)

 Terms used **software management**

 Found **94,860** of **198,617**

Sort results by


[Save results to a Binder](#)
[Try an Advanced Search](#)
[Try this search in The ACM Guide](#)

Display results


[Search Tips](#)
☐ Open results in a new window

Results 1 - 20 of 200

 Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

Best 200 shown

 Relevance scale ☐ ☐ ☐ ☐ ☐

1 [A software agent framework for the support of software project management](#)

Rita Nienaber, Elsabe Cloete

 September 2003 **Proceedings of the 2003 annual research conference of the South African institute of computer scientists and information technologists on Enablement through technology SAICSIT '03**

Publisher: South African Institute for Computer Scientists and Information Technologists

 Full text available: [pdf\(80.53 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Numerous software development projects do not live up to expectations or sadly fail. This can be seen in the fact that software projects often do not comply with the traditional standard measurements of success, namely time, cost and specifications. Traditionally, individual software projects were executed at a single location. However, due to globalisation and advances in computing technologies, this has changed, and software projects are developed and deployed in distributed and collaborative ...

Keywords: collaborative distributed software projects, design, experimentation, management, software agent computing, software project management

2 [Curriculum and content: A case for the study of software management within a broad information technology curriculum](#)

Daniel Shoemaker, Vladan Jovanovic, Antonio Drommi

 October 2003 **Proceedings of the 4th conference on Information technology curriculum CITC4 '03**

Publisher: ACM Press

 Full text available: [pdf\(193.16 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

This paper proposes the study of Software Management (SM) within a broader educational area of IT. SM focuses on building governance and technical support infrastructures rather than development of technical artifacts. In this article we compare curricular recommendations for each discipline and use these to illustrate SM's unique role.

Keywords: curriculum, information technology, software management

3 [A spiral approach to software engineering project management education](#)

Joseph C. Spicer

 March 1984 **Proceedings of the 7th international conference on Software engineering ICSE '84**

Publisher: IEEE Press

 Full text available: [pdf\(599.43 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

This paper describes the experiences of an instructor, experienced in management, teaching a software engineering project management course at a school that specializes in software education. A "spiral approach" was used to provide for the parallel acquisition of management knowledge and experience, while building on recently acquired skills in software technology and developing confidence through the successful development of a software product. This paper describes the reason for selectin ...

4 Research papers: information security and risk management: Controlling software project risks: an empirical study of methods used by experienced project managers

Tom Addison, Seema Vallabh

September 2002 **Proceedings of the 2002 annual research conference of the South African institute of computer scientists and information technologists on Enablement through technology SAICSIT '02**

Publisher: South African Institute for Computer Scientists and Information Technologists

Full text available:  [pdf\(181.20 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

The failure rate of software projects has been proven to be very high, and the incidence of failure is becoming worse as more companies venture into software development. Risk management is a collection of methods aimed at minimising or reducing the effects of project failure. This research report has focused on experienced project manager's perceptions of software project risks and controls. It reports on the more significant risks and controls that are utilised to reduce the occurrence of the ...


Keywords: control methods, management, risks

5 Research papers: information security and risk management: An interpretive study of software risk management perspectives

Keshnee Padayachee

September 2002 **Proceedings of the 2002 annual research conference of the South African institute of computer scientists and information technologists on Enablement through technology SAICSIT '02**

Publisher: South African Institute for Computer Scientists and Information Technologists

Full text available:  [pdf\(175.03 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

This research proposes a framework for a field investigation of risk management in the context of a particular software development organization. It was experimentally tested within several companies. This framework was designed to provide an understanding of software development risk phenomena from a project manager's perspective and to give an indication of how this perspective affects their perception. This study can be used as a precursor to improving research into the creation of new softwa ...

Keywords: interpretive research, phenomenological research, software risk management, software risks

6 Theory-W software project management: a case study

B. Boehm, R. Ross

April 1988 **Proceedings of the 10th international conference on Software engineering ICSE '88**

Publisher: IEEE Computer Society Press

Full text available:  [pdf\(1.24 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

The search for a single unifying principle to guide software project management has been relatively unrewarding to date. Most candidate principles are either insufficiently general to apply in many situations, or so general that they provide no useful specific guidance. This paper presents a candidate unifying principle which appears to do somewhat better. Reflecting various alphabetical management theories (X, Y, Z), it is called the Theory W approach to software project managem ...

7 Knowledge capability and maturity in software management



Richard Baskerville, Jan Pries-Heje

March 1999 **ACM SIGMIS Database**, Volume 30 Issue 2

Publisher: ACM Press

Full text available: [pdf\(1.62 MB\)](#) Additional Information: [full citation](#), [abstract](#), [citations](#)

Knowledge management is used as the underlying theory to develop a set of key process areas for a supplement to the CMM in small or medium-sized enterprises (SME) that develop software. These processes involve a focus on managing knowledge capability rather than traditional project management. A longitudinal case study provides evidence that current practices have already established the feasibility of these key process areas.

8 Version models for software configuration management



Reidar Conradi, Bernhard Westfechtel

June 1998 **ACM Computing Surveys (CSUR)**, Volume 30 Issue 2

Publisher: ACM Press

Full text available: [pdf\(483.54 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

After more than 20 years of research and practice in software configuration management (SCM), constructing consistent configurations of versioned software products still remains a challenge. This article focuses on the version models underlying both commercial systems and research prototypes. It provides an overview and classification of different versioning paradigms and defines and relates fundamental concepts such as revisions, variants, configurations, and changes. In particular, we foc ...

Keywords: changes, configuration rules, configurations, revisions, variants, versions

9 Economics, management and mathematics: Strategizing software development: strategic management of internet service development



Masao Kakiyama

May 2006 **Proceedings of the 2006 international workshop on Workshop on interdisciplinary software engineering research WISER '06**

Publisher: ACM Press

Full text available: [pdf\(392.07 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

This paper explores a theoretical linkage between software engineering and strategic management. Software engineering is now faced with two dynamic innovation streams: technological innovation and market innovation. Harshly shook by rapid technological development and highly volatile market environments, today's software development is under the constant necessity for swift and reliable development practices and market launch in appropriate timing. In short, software development has to be more a ...

Keywords: internet service, software development, strategic management

10 A variability management process for software product lines

Edson Alves de Oliveira, Itana M. S. Gimenes, Elisa Hatsue Moriya Huzita, José Carlos Maldonado

October 2005 **Proceedings of the 2005 conference of the Centre for Advanced Studies on Collaborative research CASCON '05**

Publisher: IBM Press

Full text available: [pdf\(402.68 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

The software product line approach (PL) promotes the generation of specific products from a set of core assets for a given domain. This approach is applicable to domains in which products have well-defined commonalities and variation points. Variability management is concerned with the management of the differences between products throughout the PL

lifecycle. This paper presents a UML-based process for variability management that allows identification, representation and delimitation of variability ...

11 Impact of software engineering research on the practice of software configuration management



Jacky Estublier, David Leblang, André van der Hoek, Reidar Conradi, Geoffrey Clemm, Walter Tichy, Darcy Wiborg-Weber

October 2005 **ACM Transactions on Software Engineering and Methodology (TOSEM)**, Volume 14 Issue 4

Publisher: ACM Press

Full text available: [pdf\(350.59 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Software Configuration Management (SCM) is an important discipline in professional software development and maintenance. The importance of SCM has increased as programs have become larger, more long lasting, and more mission and life critical. This article discusses the evolution of SCM technology from the early days of software development to the present, with a particular emphasis on the impact that university and industrial research has had along the way. Based on an analysis of the publication ...

Keywords: Versioning, data model, process support, research impact, software configuration management, software engineering, workspace management

12 A spiral approach to Software Engineering Project Management Education



Joseph C. Spicer

July 1983 **ACM SIGSOFT Software Engineering Notes**, Volume 8 Issue 3

Publisher: ACM Press

Full text available: [pdf\(563.06 KB\)](#) Additional Information: [full citation](#), [references](#), [citations](#)

13 Invited papers: Impact of the research community on the field of software configuration management: summary of an impact project report



Jacky Estublier, David Leblang, Geoff Clemm, Reidar Conradi, Walter Tichy, André van der Hoek, Darcy Wiborg-Weber

September 2002 **ACM SIGSOFT Software Engineering Notes**, Volume 27 Issue 5

Publisher: ACM Press

Full text available: [pdf\(1.22 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#)

Software Configuration Management (SCM) is an important discipline in professional software development and maintenance. The importance of SCM has increased as programs have become larger and more complex and mission/life-critical. This paper discusses the evolution of SCM technology from the early days of software development to present and the impact university and industrial research has had along the way. It also includes a survey of the industrial state-of-the-practice and research directions ...

Keywords: industrial impact, software configuration management, software engineering, software quality

14 Open source and distributed software development: Negotiation and the coordination of information and activity in distributed software problem management



Robert J. Sandusky, Les Gasser

November 2005 **Proceedings of the 2005 international ACM SIGGROUP conference on Supporting group work GROUP '05**

Publisher: ACM Press

Full text available: [pdf\(275.07 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index](#)

terms

Publicly accessible bug report repositories maintained by free / open source development communities provide vast stores of data about distributed software problem management (SWPM). Qualitative analysis of individual bug reports, texts that record community responses to reported software problems, shows how this distributed community uses its SWPM process to manage software quality. We focus on the role of one basic social process, *negotiation*, in SWPM. We report on the varieties and fre ...

Keywords: coordination mechanisms, coordination theory, distributed collective practices, negotiation, software problem management

15 Defining IT: Dynamic enterprises demand advanced curricula in software

development and management

Deborah G. Coleman, Stephen J. Zilora

October 2003 **Proceedings of the 4th conference on Information technology curriculum CITC4 '03**

Publisher: ACM Press

Full text available:  [pdf\(175.55 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Advancing technologies, emergent software development approaches, and economic conditions influencing corporate budgets are creating new challenges for the software development manager. Information dependent enterprises continue to require high quality software that quickly satisfies customer needs, at a time when traditional software development methods are being challenged by agile, user centric approaches. This paper focuses on one higher education program that addresses the unique needs of t ...


Keywords: degree program, design, education, programming, project management, software, software development

16 Experience with application of modern software management controls

Donald L. Paster

March 1981 **Proceedings of the 5th international conference on Software engineering ICSE '81**

Publisher: IEEE Press

Full text available:  [pdf\(910.87 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

This paper presents the experience of the Software Development Laboratory of Raytheon Company, Submarine Signal Division, in applying modern software management control techniques to the development of software for real-time embedded computer systems. The paper initially describes the characteristics of the software projects during the period 1969-1979, and the ultimate use of the systems. The software is developed for embedded computers of many types and for systems requiring from one to e ...

17 Object database support for a software project management environment

Lung-Chun Liu, Ellis Horowitz

November 1988 **ACM SIGSOFT Software Engineering Notes , ACM SIGPLAN Notices , Proceedings of the third ACM SIGSOFT/SIGPLAN software engineering symposium on Practical software development environments SDE 3, Volume 13 , 24 Issue 5 , 2**

Publisher: ACM Press

Full text available:  [pdf\(1.39 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

The recent development of object-oriented database models, which combine the power of object programming and the efficient management of data, provides a feasible solution for the construction of a computer-aided software engineering environment or CASE. However, an object oriented database provides only a kernel set of capabilities. This paper identifies the data management requirements related to software project management

and shows how they are represented in the model called Design-Net ...

18 Software configuration management: a roadmap



Jacky Estublier

May 2000 **Proceedings of the Conference on The Future of Software Engineering ICSE '00**

Publisher: ACM Press

Full text available: [pdf\(925.61 KB\)](#) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

Keywords: architecture, concurrent engineering, federation, interoperability, process support, software configuration management, version control

19 Coordination: Software configuration management over a global software development environment: lessons learned from a case study



Leonardo Pilatti, Jorge Luis Nicolas Audy, Rafael Prikladnicki

May 2006 **Proceedings of the 2006 international workshop on Global software development for the practitioner GSD '06**

Publisher: ACM Press

Full text available: [pdf\(234.54 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Software configuration management is an important support activity in the software development process. In global environments, the software configuration becomes critical due to the characteristics of the distributed development (physical distance, cultural differences, trust, communication and other factors). The objective of this paper is to analyze the software configuration management in a global software development environment, identifying the main challenges. The results are based on a c ...

Keywords: global software development, software configuration management, software process improvement

20 Emerging results: metrics: Portfolio management of software development projects using COCOMO II



Wiboon Jiamthubhugsin, Daricha Sutivong

May 2006 **Proceeding of the 28th international conference on Software engineering ICSE '06**

Publisher: ACM Press

Full text available: [pdf\(130.10 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Software development projects are subject to external and internal risks that cause delays, budget overrun and poor quality. Portfolio management can be used to alleviate this problem, as it pools resources together and allows for resource sharing among projects. Consequently, projects are more likely to succeed. However, portfolio management using only deadlines and the number of employees to improve probability of success is still confined. This paper proposes integrating portfolio management ...

Keywords: COCOMO II, portfolio management, software development, software project, software project risk

Results 1 - 20 of 200

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2007 ACM, Inc.

[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Contact Us](#)

Useful downloads: [Adobe Acrobat](#) [QuickTime](#) [Windows Media Player](#) [Real Player](#)


[Subscribe \(Full Service\)](#) [Register \(Limited Service, Free\)](#) [Login](#)

 Search: ☒ The ACM Digital Library ☐ The Guide

THE ACM DIGITAL LIBRARY


[Feedback](#) [Report a problem](#) [Satisfaction survey](#)

 Terms used **software upgrading**

 Found **69,635** of **198,617**

Sort results by


[Save results to a Binder](#)
[Try an Advanced Search](#)
[Try this search in The ACM Guide](#)

Display results


[Search Tips](#)
☐ Open results in a new window

Results 1 - 20 of 200

 Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

Best 200 shown

 Relevance scale ☐ ☐ ☐ ☐ ☐

1 [Special session on software systems #2: Case study: lessons learned during a nationwide computer system upgrade](#)



Maureen Ann Raley, Letha Hughes Etzkorn

 April 2004 **Proceedings of the 42nd annual Southeast regional conference ACM-SE 42**

Publisher: ACM Press

 Full text available: pdf(312.78 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

This paper is a case study that presents the results of a lessons-learned questionnaire gathered during the final six months of a nationwide computer system upgrade. The method used to capture the lessons learned feedback is presented. Common complaints and obstacles encountered during the system upgrade are summarized and possible improvements are suggested. The case study indicated that, although the project was completed on time, a more rigorous solicitation of feedback on management directio ...

Keywords: COTS, COTS-based systems, lessons learned, project management, quality control, software maintenance, system upgrade

2 [Continuous release and upgrade of component-based software](#)



Tijs van der Storm

 September 2005 **Proceedings of the 12th international workshop on Software configuration management SCM '05**

Publisher: ACM Press

 Full text available: pdf(619.47 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#)

We show how under certain assumptions, the release and delivery of software updates can be automated in the context of component-based systems. These updates allow features or fixes to be delivered to users more quickly. Furthermore, user feedback is more accurate, thus enabling quicker response to defects encountered in the field. Based on a formal product model we extend the process of continuous integration to enable the agile and automatic release of software components component. From such r ...

3 [Predicting problems caused by component upgrades](#)



Stephen McCamant, Michael D. Ernst

 September 2003 **ACM SIGSOFT Software Engineering Notes , Proceedings of the 9th European software engineering conference held jointly with 11th ACM SIGSOFT international symposium on Foundations of software engineering ESEC/FSE-11, Volume 28 Issue 5**

Publisher: ACM Press

 Full text available: pdf(142.32 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

We present a new, automatic technique to assess whether replacing a component of a software system by a purportedly compatible component may change the behavior of the system. The technique operates before integrating the new component into the system or running system tests, permitting quicker and cheaper identification of problems. It takes into account the system's use of the component, because a particular component upgrade may be desirable in one context but undesirable in another. No forma ...

Keywords: software components, software upgrades, specification matching

4 The optimal software licensing policy under quality uncertainty



Jie Zhang, Abraham Seidmann

September 2003 **Proceedings of the 5th international conference on Electronic commerce ICEC '03**

Publisher: ACM Press

Full text available: [pdf\(237.20 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#)

A new service model has emerged which delivers application software and services over the Web on a lease or subscription basis. Our paper studies the optimal licensing policy of a software vendor that uses that business model. We look at software vendors that are both selling (at a posted price) or leasing their products where as lessor they guarantee that the lessee will always have the latest version of the software on their desktop. We address some of the specific issues of implementing this ...

Keywords: network externality, price discrimination, risk, software licensing, upgrade compatibility

5 Quantitative assessment of the software maintenance process and requirements volatility



Joel Henry, Sallie Henry

March 1993 **Proceedings of the 1993 ACM conference on Computer science CSC '93**

Publisher: ACM Press

Full text available: [pdf\(769.50 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

This paper describes analysis techniques used to quantitatively assess the software maintenance process of a large military contractor, and the results obtained. The analysis techniques make use of basic data collected throughout the maintenance process. The data collected are extensive and allow a set of functional enhancements to be traced to process activities and product impact. Simple nonparametric statistical techniques are then applied to test relationships between data items ...

6 Increasing the confidence in off-the-shelf components: a software connector-based approach



Marija Rakic, Nenad Medvidovic

May 2001 **ACM SIGSOFT Software Engineering Notes , Proceedings of the 2001 symposium on Software reusability: putting software reuse in context SSR '01**, Volume 26 Issue 3

Publisher: ACM Press

Full text available: [pdf\(759.03 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

The promise of component-based software development is that larger, more complex systems can be built reasonably quickly and reliably from pre-fabricated (" off-the-shelf") building blocks. Additionally, such systems can be upgraded incrementally, simply by replacing individual components with their new versions. However, practice has shown that while it may improve certain aspects of an existing component, a new component version frequently introduces unforeseen problems. These ...

Windows 2000 lab system upgrade: good idea, bad idea?

Jesse Ballard, James Osborn

October 2000 **Proceedings of the 28th annual ACM SIGUCCS conference on User services: Building the future SIGUCCS '00**

Publisher: ACM Press

Full text available: [pdf\(130.68 KB\)](#) Additional Information: [full citation](#), [index terms](#)**Keywords:** Windows 2000, lab system, operating systems, upgrade

8

Approaches to upgrading software process maturity

Judah Mogilensky

July 1990 **Proceedings of the seventh Washington Ada symposium on Ada WADAS '90**

Publisher: ACM Press

Full text available: [pdf\(527.81 KB\)](#) Additional Information: [full citation](#), [references](#), [index terms](#)

9

Software quality and process: Predictors of customer perceived software quality

Audris Mockus, Ping Zhang, Paul Luo Li

May 2005 **Proceedings of the 27th international conference on Software engineering ICSE '05 , Proceedings of the 27th international conference on Software engineering ICSE '05**

Publisher: ACM Press, IEEE Computer Society

Full text available: [pdf\(107.20 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)
 [Publisher Site](#)

Predicting software quality as perceived by a customer may allow an organization to adjust deployment to meet the quality expectations of its customers, to allocate the appropriate amount of maintenance resources, and to direct quality improvement efforts to maximize the return on investment. However, customer perceived quality may be affected not simply by the software content and the development process, but also by a number of other factors including deployment issues, amount of usage, softwa ...

Keywords: metrics, modeling, quality

10

ERP II: best practices for successfully implementing an ERP upgrade

Robert C. Beatty, Craig D. Williams

March 2006 **Communications of the ACM**, Volume 49 Issue 3

Publisher: ACM Press

Full text available: [pdf\(82.43 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)
 [html\(25.88 KB\)](#)

An ERP upgrade is deceptively complex and can be daunting---especially for organizations ignorant of the massive effort required to do it correctly.

11

Experience papers: software development practices: Applying the Value/Petri process to ERP software development in China

LiGuo Huang, Barry Boehm, Hao Hu, Jidong Ge, Jian Lü, Cheng Qian

May 2006 **Proceeding of the 28th international conference on Software engineering ICSE '06**

Publisher: ACM Press

Full text available: [pdf\(4.41 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Commercial organizations increasingly need software processes sensitive to business value, quick to apply, and capable of early analysis for subprocess consistency and

compatibility. This paper presents experience in applying a lightweight synthesis of a Value-Based Software Quality Achievement (VBSQA) process and an Object-Petri-Net-based process model (called VBSQA-OPN) to achieve a manager-satisfactory process for software quality achievement in an on-going ERP software project in China. The ...

Keywords: ROI, cost, object petri nets (OPN), software process formalization, software process simulation, software quality, value

12 Session 6A: Architectures: Practical perspectives on software architectures, high-level design, and evolution



Tommi Mikkonen, Peeter Pruuden

September 2001 **Proceedings of the 4th International Workshop on Principles of Software Evolution IWPSE '01**

Publisher: ACM Press

Full text available: [pdf\(408.17 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Evolving software gets more complex in each increment. As real-life increments tend to be additive rather than upgrades with more fundamental purpose, the underlying code base keeps extending. With such increments, the associated core architecture of the system gets more and more difficult to modify, because an increasing number of functions are attached to it. Therefore, only the first versions of systems can be properly architected, whereas later versions rely on an already existing architecture ...

Keywords: high-level design, software architecture, software evolution

13 Practical dynamic software updating for C



Iulian Neamtiu, Michael Hicks, Gareth Stoye, Manuel Oriol

June 2006 **ACM SIGPLAN Notices , Proceedings of the 2006 ACM SIGPLAN conference on Programming language design and implementation PLDI '06**, Volume 41 Issue 6

Publisher: ACM Press

Full text available: [pdf\(525.67 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Software updates typically require stopping and restarting an application, but many systems cannot afford to halt service, or would prefer not to. *Dynamic software updating* (DSU) addresses this difficulty by permitting programs to be updated while they run. DSU is appealing compared to other approaches for on-line upgrades because it is quite general and requires no redundant hardware. The challenge is in making DSU *practical*: it should be flexible, and yet safe, efficient, and eas ...

Keywords: dynamic software updating, function indirection, loop extraction, type wrapping

14 Techniques for trusted software engineering

Premkumar T. Devanbu, Philip W-L Fong, Stuart G. Stubblebine

April 1998 **Proceedings of the 20th international conference on Software engineering ICSE '98**

Publisher: IEEE Computer Society

Full text available: [pdf\(1.21 MB\)](#) [Publisher Site](#) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

15 Dynamic software updating



Michael Hicks, Scott Nettles

November 2005 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 27 Issue 6

Publisher: ACM Press

Full text available:  pdf(622.69 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Many important applications must run continuously and without interruption, and yet also must be changed to fix bugs or upgrade functionality. No prior general-purpose methodology for dynamic updating achieves a practical balance between flexibility, robustness, low overhead, ease of use, and low cost. We present an approach for C-like languages that provides type-safe dynamic updating of native code in an extremely flexible manner---code, data, and types may be updated, at programmer-determined ...

Keywords: Dynamic software updating, typed assembly language

16 Hardware and software choices for student computer initiatives



Richard J. LeBlanc, Steven L. Teal

January 1998 **Communications of the ACM**, Volume 41 Issue 1

Publisher: ACM Press

Full text available:  pdf(96.47 KB) Additional Information: [full citation](#), [index terms](#), [review](#)

17 Software operation and evolution: Empirical estimates of software availability of deployed systems



Audris Mockus

September 2006 **Proceedings of the 2006 ACM/IEEE international symposium on International symposium on empirical software engineering ISESE '06**

Publisher: ACM Press

Full text available:  pdf(313.65 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

We consider empirical evaluation of the availability of the deployed software. Evaluation of real systems is more realistic, more accurate, and provides higher level of confidence than simulations, testing, or models. We process and model information gathered from a variety of operational and service support systems to obtain estimates of software reliability and availability. The three principal quantities are the total runtime, the number of outages, and the duration of outages. We consider me ...

18 Lessons learned from the novell and groupWise upgrade of the summer 2003 or "what we did on our summer vacation!"



Robert L. Barley, Yancy Phillips

October 2004 **Proceedings of the 32nd annual ACM SIGUCCS conference on User services SIGUCCS '04**

Publisher: ACM Press

Full text available:  pdf(174.72 KB) Additional Information: [full citation](#), [abstract](#), [index terms](#)

This presentation will focus on the effects a recent major hardware and software upgrade had on the Indiana State University end user community and how collaboration between Technical and User Services resulted in a positive experience for the end user community. There were many hours of overtime put into this project and lots of concerns from User Services personnel about having to go out and "touch" each computer to facilitate necessary changes. The effects of the unsolicited visits produce ...

Keywords: novell upgrades, reliability, standardization, support

19 Evolutionary design of complex software (EDCS) demonstration days 1999



Wayne Stidolph

January 2000 **ACM SIGSOFT Software Engineering Notes**, Volume 25 Issue 1

Publisher: ACM Press

Full text available:  [pdf\(1.90 MB\)](#) Additional Information: [full citation](#), [abstract](#), [index terms](#)

This report summarizes the Product/Technology demonstrations given at Defense Advanced Research Projects Agency (DARPA) Evolutionary Design of Complex Software (EDCS) Program Demonstration Days, held 28-29 June 1999 at the Sheraton National Hotel, Arlington, VA.

20 Twenty dirty tricks to train software engineers



Ray Dawson

June 2000 **Proceedings of the 22nd international conference on Software engineering ICSE '00**

Publisher: ACM Press

Full text available:  [pdf\(119.44 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Many employers find that graduates and sandwich students come to them poorly prepared for the every day problems encountered at the workplace. Although many university students undertake team projects at their institutions, an education environment has limitations that prevent the participants experiencing the full range of problems encountered in the real world. To overcome this, action was taken on courses at the Plessey Telecommunications company and Loughborough University to disrupt th ...

Keywords: education, project, teamwork, training, work experience

Results 1 - 20 of 200

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2007 ACM, Inc.

[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Contact Us](#)

Useful downloads:  [Adobe Acrobat](#)  [QuickTime](#)  [Windows Media Player](#)  [Real Player](#)



USPTO

[Subscribe \(Full Service\)](#) [Register \(Limited Service, Free\)](#) [Login](#)

 Search: ☒ The ACM Digital Library ☐ The Guide

software updating

SEARCH

THE ACM DIGITAL LIBRARY


[Feedback](#) [Report a problem](#) [Satisfaction survey](#)
Terms used **software updating**Found **88,659** of **198,617**

Sort results by

relevance

[Save results to a Binder](#)Try an [Advanced Search](#)

Display results

expanded form

[Search Tips](#)Try this search in [The ACM Guide](#)
☐ Open results in a new window

Results 1 - 20 of 200

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

Best 200 shown

Relevance scale ☐ ☐ ☐ ☐ ☐

1 [Sensor network software update management: a survey](#)

Chih-Chieh Han, Ram Kumar, Roy Shea, Mani Srivastava

July 2005 **International Journal of Network Management**; Volume 15 Issue 4

Publisher: John Wiley & Sons, Inc.

Full text available: pdf(158.17 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Software management is a critical task in the system administration of enterprise-scale networks. Enterprise-scale networks that have traditionally comprised of large clusters of workstations are expanding to include low-power ad hoc wireless sensor networks (WSN). The existing tools for software updates in workstations cannot be used with the severely resource-constrained sensor nodes. In this article, we survey the software update techniques in WSNs. We base our discussion around a conceptual ...

2 [Dynamic software updating](#)



Michael Hicks, Scott Nettles

November 2005 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 27 Issue 6

Publisher: ACM Press

Full text available: pdf(622.69 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Many important applications must run continuously and without interruption, and yet also must be changed to fix bugs or upgrade functionality. No prior general-purpose methodology for dynamic updating achieves a practical balance between flexibility, robustness, low overhead, ease of use, and low cost. We present an approach for C-like languages that provides type-safe dynamic updating of native code in an extremely flexible manner---code, data, and types may be updated, at programmer-determined ...

Keywords: Dynamic software updating, typed assembly language

3 [A software package for sparse orthogonal factorization and updating](#)



Ove Edlund

December 2002 **ACM Transactions on Mathematical Software (TOMS)**, Volume 28 Issue 4

Publisher: ACM Press

Full text available: pdf(490.01 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Although there is good software for sparse QR factorization, there is little support for updating and downdating, something that is absolutely essential in some linear programming algorithms, for example. This article describes an implementation of sparse LQ factorization, including block triangularization, approximate minimum degree ordering, symbolic factorization, multifrontal factorization, and updating and downdating. The factor

Q is not retained. The updating algorithm expands the n ...

Keywords: Sparse matrix, downdating, orthogonal factorization, software, updating

4 Planet scale software updates

 Christos Gkantsidis, Thomas Karagiannis, Milan Vojnovic
August 2006 **ACM SIGCOMM Computer Communication Review , Proceedings of the 2006 conference on Applications, technologies, architectures, and protocols for computer communications SIGCOMM '06**, Volume 36 Issue 4


Publisher: ACM Press

Full text available:  [pdf\(1.19 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Fast and effective distribution of software updates (a.k.a. patches) to millions of Internet users has evolved into a critical task over the last years. In this paper, we characterize "Windows Update", one of the largest update services in the world, with the aim to draw general guidelines on how to best design and architect a fast and effective planet-scale patch dissemination system. To this end, we analyze an extensive set of data traces collected over the period of a year, consisting of bill ...

Keywords: caching, peer-to-peer, software updates

5 Practical dynamic software updating for C

 Iulian Neamtiu, Michael Hicks, Gareth Stoye, Manuel Oriol
June 2006 **ACM SIGPLAN Notices , Proceedings of the 2006 ACM SIGPLAN conference on Programming language design and implementation PLDI '06**, Volume 41 Issue 6


Publisher: ACM Press

Full text available:  [pdf\(525.67 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Software updates typically require stopping and restarting an application, but many systems cannot afford to halt service, or would prefer not to. *Dynamic software updating* (DSU) addresses this difficulty by permitting programs to be updated while they run. DSU is appealing compared to other approaches for on-line upgrades because it is quite general and requires no redundant hardware. The challenge is in making DSU *practical*: it should be flexible, and yet safe, efficient, and eas ...

Keywords: dynamic software updating, function indirection, loop extraction, type wrapping

6 Short papers: Evaluating the release, delivery, and deployment processes of eight large product software vendors applying the customer configuration update model

 Slinger Jansen, Sjaak Brinkkemper
May 2006 **Proceedings of the 2006 international workshop on Workshop on interdisciplinary software engineering research WISER '06**

Publisher: ACM Press

Full text available:  [pdf\(68.61 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)


For software vendors the processes of release, delivery, and deployment to customers are inherently complex. However, software vendors can greatly improve their product quality and quality of service by applying a model that focuses on customer interaction if such a model were available. This paper presents a model for customer configuration updating (CCU) that can evaluate the capabilities of a software vendor in these processes. Eight extensive case studies of medium to large product software ...

7 Dynamic software updating

 Michael Hicks, Jonathan T. Moore, Scott Nettles
May 2001 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 2001 conference**

on Programming language design and implementation PLDI '01, Volume 36

Issue 5

Publisher: ACM PressFull text available:  [pdf\(1.44 MB\)](#)Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Many important applications must run continuously and without interruption, yet must be changed to fix bugs or upgrade functionality. No prior general-purpose methodology for dynamic updating achieves a practical balance between flexibility, robustness, low overhead, and ease of use.

We present a new approach for C-like languages that provides type-safe dynamic updating of native code in an extremely flexible manner (code, data, and types may be updated, at programmer-determined times ...

8 Doctoral symposium: presentations: Improving the customer configuration update process by explicitly managing software knowledge



Slinger Jansen

May 2006 **Proceeding of the 28th international conference on Software engineering ICSE '06****Publisher:** ACM PressFull text available:  [pdf\(77.84 KB\)](#)Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

The implementation and continuous support of a software product at a customer with evolving requirements is a complex task for a product software vendor. There are many customers for the vendor to serve, all of whom might require their own version or variant of the application. Furthermore, the software application itself will consist of many (software) components that depend on each other to function correctly. On top of that, these components will evolve over time to meet the changing needs of ...

9 Mutatis mutandis: safe and predictable dynamic software updating



Gareth Stoye, Michael Hicks, Gavin Bierman, Peter Sewell, Iulian Neamtii

January 2005 **ACM SIGPLAN Notices , Proceedings of the 32nd ACM SIGPLAN-SIGACT symposium on Principles of programming languages POPL '05**, Volume 40

Issue 1

Publisher: ACM PressFull text available:  [pdf\(273.03 KB\)](#)Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Dynamic software updates can be used to fix bugs or add features to a running program without downtime. Essential for some applications and convenient for others, low-level dynamic updating has been used for many years. Perhaps surprisingly, there is little high-level understanding or language support to help programmers write dynamic updates effectively. To bridge this gap, we present Proteus, a core calculus for dynamic software updating in C-like languages that is flexible, safe, and predictable ...

Keywords: capability, dynamic software updating, proteus, type inference, updateability analysis

10 Agents, interactions, mobility and systems: Software update via mobile agent based programming



Lorenzo Bettini, Rocco De Nicola, Michele Loreti

March 2002 **Proceedings of the 2002 ACM symposium on Applied computing SAC '02****Publisher:** ACM PressFull text available:  [pdf\(534.28 KB\)](#)Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

We describe a system that permits maintaining the software installed on several heterogeneous computers distributed over a network by taking advantage of the mobile agent paradigm. The applications are installed and updated only on the central server.

When a new release of an application is installed on the server, agents are scattered along the network to update the application on the clients. To build a prototype system we use X-KLAIM, a programming language specifically designed to pr ...

Keywords: distributed software update, mobile agents, mobile code

11 From the editors: An update on software updates



David J. Brown

March 2005 **Queue**, Volume 3 Issue 2

Publisher: ACM Press

Full text available: pdf(82.27 KB)

html(8.08 KB)

Additional Information: [full citation](#), [abstract](#), [citations](#), [index terms](#)

The way software is delivered has changed.

12 A rule-based language for programming software updates



Martin Erwig, Deling Ren

October 2002 **Proceedings of the 2002 ACM SIGPLAN workshop on Rule-based programming RULE '02**

Publisher: ACM Press

Full text available: pdf(118.18 KB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

We describe the design of a rule-based language for expressing changes to Haskell programs in a systematic and reliable way. The update language essentially offers update commands for all constructs of the object language (a subset of Haskell). The update language can be translated into a core calculus consisting of a small set of basic updates and update combinators. The key construct of the core calculus is a scope update mechanism that allows (and enforces) update specifications for the defin ...

Keywords: type change, type correctness, update program, update safety

13 PLI workshops: A rule-based language for programming software updates



Martin Erwig, Deling Ren

December 2002 **ACM SIGPLAN Notices**, Volume 37 Issue 12

Publisher: ACM Press

Full text available: pdf(182.21 KB)

Additional Information: [full citation](#), [abstract](#), [references](#)

We describe the design of a rule-based language for expressing changes to Haskell programs in a systematic and reliable way. The update language essentially offers update commands for all constructs of the object language (a subset of Haskell). The update language can be translated into a core calculus consisting of a small set of basic updates and update combinators. The key construct of the core calculus is a scope update mechanism that allows (and enforces) update specifications for the defin ...

Keywords: Haskell, type change, type correctness, update program, update safety

14 Deploying, updating, and managing tools for collecting software metrics



Alberto Sillitti, Barbara Russo, Paolo Zuliani, Giancarlo Succi

November 2004 **Proceedings of the 2004 workshop on Quantitative techniques for software agile process QUTE-SWAP '04**

Publisher: ACM Press

Full text available: pdf(351.02 KB)

Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Collecting software engineering data is difficult due to the number of problems that researchers face in this activity. One of the most relevant is the ability to install and keep up-to-date the measurement tools installed in the production machines in order to collect

such data. Even when a few machines are involved, maintaining all the tools required to collect data from the different development tools requires a full-time system administrator. Moreover, since most of these tools are research ...

Keywords: process monitoring, software metrics

15 Full papers: Tree bitmap: hardware/software IP lookups with incremental updates



Will Eatherton, George Varghese, Zubin Dittia

April 2004 **ACM SIGCOMM Computer Communication Review**, Volume 34 Issue 2

Publisher: ACM Press

Full text available: [pdf\(189.39 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#)

Even with the significant focus on IP address lookup in the published literature as well as focus on this market by commercial semiconductor vendors, there is still a challenge for router architects to find solutions that simultaneously meet 3 criteria: scaling in terms of lookup speeds as well as table sizes, the ability to perform high speed updates, and the ability to fit into the overall memory architecture of an Level 3 forwarding engine or packet processor with low systems cost overhead. I ...

16 Towards a unified formal model for supporting mechanisms of dynamic component update



Junrong Shen, Xi Sun, Gang Huang, Wenpin Jiao, Yanchun Sun, Hong Mei

September 2005 **ACM SIGSOFT Software Engineering Notes , Proceedings of the 10th European software engineering conference held jointly with 13th ACM SIGSOFT international symposium on Foundations of software engineering ESEC/FSE-13**, Volume 30 Issue 5

Publisher: ACM Press

Full text available: [pdf\(268.30 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

The continuous requirements of evolving a delivered software system and the rising cost of shutting down a running software system are forcing researchers and practitioners to find ways of updating software as it runs. Dynamic update is a kind of software evolution that updates a running program without interruption. This paper covers the fundamental issues of the mechanisms of dynamic update theoretically. Based on a similarity analysis of many typical approaches to dynamic update during the pa ...

Keywords: CSP, architectural connector, dynamic update, software architecture

17 A cooperative approach to support software deployment using the software dock

Richard S. Hall, Dennis Heimbigner, Alexander L. Wolf

May 1999 **Proceedings of the 21st international conference on Software engineering ICSE '99**

Publisher: IEEE Computer Society Press

Full text available: [pdf\(1.43 MB\)](#) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

Keywords: Java, configuration management, mobile agents, software deployment

18 Continuous release and upgrade of component-based software



Tijs van der Storm

September 2005 **Proceedings of the 12th international workshop on Software configuration management SCM '05**

Publisher: ACM Press

Full text available: [pdf\(619.47 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#)

We show how under certain assumptions, the release and delivery of software updates can be automated in the context of component-based systems. These updates allow features or fixes to be delivered to users more quickly. Furthermore, user feedback is more accurate, thus enabling quicker response to defects encountered in the field. Based on a formal product model we extend the process of continuous integration to enable the agile and automatic release of software components component. From such r ...

19 Applying data mining to software maintenance records

Jelber Sayyad Shirabad, Timothy C. Lethbridge, Stan Matwin

October 2003 **Proceedings of the 2003 conference of the Centre for Advanced Studies on Collaborative research CASCON '03**

Publisher: IBM Press

Full text available:  [pdf\(140.30 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

In a system maintained over a long time period, as is the case for legacy software, there will be many unknown and non-trivial relationships among components. Finding such hidden relationships may help software engineers in their maintenance activities. In this paper we present an approach whereby we mine software update records to find relationships between files that are changed together. The generalized models we present as results are obtained by using features extracted from different sourc ...

20 Software engineering: sound solutions for the 21st century: Supporting transparent model update in distributed CASE tool integration

Prawee Sriplakich, Xavier Blanc, Marie-Pierre Gervais

April 2006 **Proceedings of the 2006 ACM symposium on Applied computing SAC '06**

Publisher: ACM Press

Full text available:  [pdf\(851.89 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Model Driven Architecture (MDA) is a software development approach that focuses on models. In order to support MDA, a lot of CASE tools have emerged; each of them provides a different set of modeling services (operations for automating model manipulation). We have proposed an open environment called ModelBus, which enables the integration of heterogeneous and distributed CASE tools. ModelBus enables tools to invoke the modeling services provided by other tools. In this paper, we focus on support ...

Keywords: CASE tool, RPC, call-by-copy-restore, data structure, graph, integration, interoperability, middleware

Results 1 - 20 of 200

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2007 ACM, Inc.

[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Contact Us](#)

Useful downloads:  [Adobe Acrobat](#)  [QuickTime](#)  [Windows Media Player](#)  [Real Player](#)


[Subscribe \(Full Service\)](#) [Register \(Limited Service, Free\)](#) [Login](#)

 Search: ☒ The ACM Digital Library ☐ The Guide

THE ACM DIGITAL LIBRARY


[Feedback](#) [Report a problem](#) [Satisfaction survey](#)

 Terms used **software management** and **product and update**

 Found **92,083** of **198,617**

Sort results by


[Save results to a Binder](#)
[Try an Advanced Search](#)
[Try this search in The ACM Guide](#)

Display results


[Search Tips](#)
☐ Open results in a new window

Results 1 - 20 of 200

 Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

Best 200 shown

 Relevance scale ☐ ☐ ☐ ☐ ☐

- 1 [Short papers: Evaluating the release, delivery, and deployment processes of eight large product software vendors applying the customer configuration update model](#)



Slinger Jansen, Sjaak Brinkkemper

 May 2006 **Proceedings of the 2006 international workshop on Workshop on interdisciplinary software engineering research WISER '06**

Publisher: ACM Press

 Full text available: pdf(68.61 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

For software vendors the processes of release, delivery, and deployment to customers are inherently complex. However, software vendors can greatly improve their product quality and quality of service by applying a model that focuses on customer interaction if such a model were available. This paper presents a model for customer configuration updating (CCU) that can evaluate the capabilities of a software vendor in these processes. Eight extensive case studies of medium to large product software ...

- 2 [Doctoral symposium: presentations: Improving the customer configuration update process by explicitly managing software knowledge](#)



Slinger Jansen

 May 2006 **Proceeding of the 28th international conference on Software engineering ICSE '06**

Publisher: ACM Press

 Full text available: pdf(77.84 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

The implementation and continuous support of a software product at a customer with evolving requirements is a complex task for a product software vendor. There are many customers for the vendor to serve, all of whom might require their own version or variant of the application. Furthermore, the software application itself will consist of many (software) components that depend on each other to function correctly. On top of that, these components will evolve over time to meet the changing needs of ...

- 3 [Software Configuration Management](#)



Edward H. Bersoff, Vilas D. Henderson, Stan G. Siegel

 January 1978 **ACM SIGSOFT Software Engineering Notes , ACM SIGMETRICS Performance Evaluation Review , Proceedings of the software quality assurance workshop on Functional and performance issues**, Volume 3 , 7 Issue 5 , 3-4

Publisher: ACM Press

 Full text available: pdf(774.00 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

This paper is about discipline. It is about discipline that managers should apply to software development. Why is such discipline needed? Quite simply because the software industry has traditionally behaved in an undisciplined manner—doing its own thing. The

products that the industry has turned out have typically Contained other than what was expected (usually less, rather than more); Been delivered much later than scheduled; Cost more than anticipate ...

4 The critical elements of patch management



Thomas Gerace, Huseyin Cavusoglu

November 2005 **Proceedings of the 33rd annual ACM SIGUCCS conference on User services SIGUCCS '05**

Publisher: ACM Press

Full text available: pdf(166.10 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Only a few years ago, the term "patch management" was not in the general vocabulary of even the most advanced information technology staff. Today, "patch management" is not only in the general vernacular of most IT staff, but it is also one of the more essential responsibilities of IT departments. Security threats stemming from the exploitation of vulnerabilities in software products pose an important problem to corporations, governmental agencies, educational institutions, banking, and many oth ...

Keywords: Microsoft, automatic update, operating system, patch, patch management, security, update, windows

5 Transition to object-oriented software development



Mohamed E. Fayad, Wei-Tek Tsai, Milton L. Fulghum

February 1996 **Communications of the ACM**, Volume 39 Issue 2

Publisher: ACM Press

Full text available: pdf(451.14 KB) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

6 Software Manufacturing

L. Bernstein, C. M. Yuh

June 1978 **Proceedings of the 15th conference on Design automation DAC '78**

Publisher: IEEE Press

Full text available: pdf(615.71 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

This paper suggests a unique method of organizing and staffing for the production of deliverable computer-based systems which takes advantage of assembly-line techniques. Although these production activities are common to most projects, Software Manufacturing is defined here as an inline, rather than support, function requiring special skills. A system development cycle is described, detailing the Software Manufacturing tasks with considerations for introducing this functions into existing ...

7 The challenges and successes of implementing an enterprise patch management solution



Tom Gerace, Jean Mouton

October 2004 **Proceedings of the 32nd annual ACM SIGUCCS conference on User services SIGUCCS '04**

Publisher: ACM Press

Full text available: pdf(138.87 KB) Additional Information: [full citation](#), [abstract](#), [index terms](#)


With the proliferation of security threats we are required today, more than ever before, to effectively manage computer systems. Computers that we are responsible for in labs, classrooms, and faculty and staff offices require constant attention as manufacturer-supplied program patches and updates become available. Missing a critical update or patch in your networked environment can spell disaster or cost countless hours of "cleaning up" after an errant worm wreaks havoc on unpatched machines. ...

Keywords: HFNetChk, HFNetChkPro, SUS, automatic update, microsoft, operating system, patch, security, update

8 Configuration control for evolutionary software products

Osamu Shigo, Yoshio Wada, Yuichi Terashima, Kanji Iwamoto, Takashi Nishimura
September 1982 **Proceedings of the 6th international conference on Software engineering ICSE '82**

Publisher: IEEE Computer Society Press

Full text available:  [pdf\(600.05 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

This paper describes the concept of and a system for configuration control for evolutionary software products, in which a wide spectrum of varied software products are being continuously evolved, along with rapid advancements in hardware technologies. The system contains a database for dealing with the overall configuration structure, including hierarchical product structure with change status, master file directories, difficulty occurrences and user information. The data representing the co ...

9 Microcomputer software distribution: valuable or valueless exercise in resource management?

Lawrence A. Pounds
September 1991 **Proceedings of the 19th annual ACM SIGUCCS conference on User services SIGUCCS '91**


Publisher: ACM Press

Full text available:  [pdf\(552.37 KB\)](#) Additional Information: [full citation](#), [citations](#), [index terms](#)

10 The computer assisted software engineering (CASE) system

William Scott Amey
September 1979 **Proceedings of the 4th international conference on Software engineering ICSE '79**

Publisher: IEEE Press


Full text available:  [pdf\(398.18 KB\)](#) Additional Information: [full citation](#), [abstract](#), [index terms](#)

The CASE system provides a comprehensive software development tool that aids engineers, designers, programmers, and managers. The CASE system guides its users through the tasks of documentation, design, coding, testing, configuration control, and status analysis.

11 Web-based tools, systems and environments: Software configuration, distribution, and deployment of web-services

Rainer Anzböck, Schahram Dustdar, Harald Gall
July 2002 **Proceedings of the 14th international conference on Software engineering and knowledge engineering SEKE '02**

Publisher: ACM Press

Full text available:  [pdf\(519.92 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Web-Services can be seen as a newly emerging distributed computing model for the Web. They cater for the need to establish business-to-business (B2B) interactions on the Web. Web-Services consider a loosely coupled component model encapsulating business logic and interact with other components using XML protocols. Based on one case study, this paper discusses architectural issues and requirements for software configuration, distribution, and deployment of web-services.

Keywords: software architecture, software distribution environments, web-services

12 Security III: Wireless security patch management system

Charles Higby, Michael Bailey



October 2004 **Proceedings of the 5th conference on Information technology education**
CITC5 '04

Publisher: ACM Press

Full text available: pdf(526.37 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Wireless access on college campuses facilitates the spread of computer viruses and worms due to laptops that do not have current software patches and/or antivirus protection connecting to the network.

Wireless local area networks provide a luxury of mobility to clients so that they may roam about without the restriction of wires, room, and/or buildings. At the same time, wireless local area networks have given birth to a new breed of network weaknesses that are compounding and exploiti ...

Keywords: administration, configuration, patch, security, update, virus, wireless, worm

13 Why is software always late?



J. L. Lawrence

January 1985 **ACM SIGSOFT Software Engineering Notes**, Volume 10 Issue 1

Publisher: ACM Press

Full text available: pdf(1.03 MB) Additional Information: [full citation](#), [abstract](#)

Despite all of the advances in software engineering practice, despite all the newly developed languages and software tools, and despite case study after case study, software is almost always late. It does not seem to matter what the product is or what the industry is. The cry of frustration is almost always the same: "Why is software always late?" In this article, the author discusses the software development cycle and the LOC/day productivity measure in an attempt to explore some of the reasons ...

14 APPL/A: a language for software process programming



Stanley M. Sutton, Dennis Heimbigner, Leon J. Osterweil

July 1995 **ACM Transactions on Software Engineering and Methodology (TOSEM)**,
Volume 4 Issue 3

Publisher: ACM Press

Full text available: pdf(4.89 MB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

Software process programming is the coding of software processes in executable programming languages. Process programming offers many potential benefits, but their realization has been hampered by a lack of experience in the design and use of process programming languages. APPL/A is a prototype software process programming language developed to help gain this experience. It is intended for the coding of programs to represent and support software processes including process, product, and p ...

Keywords: consistency management, multiparadigm programming languages, software process programming, transaction management

15 Self-assessment procedure X: a self-assessment procedure dealing with software project management



Roger S. Gourd

December 1982 **Communications of the ACM**, Volume 25 Issue 12

Publisher: ACM Press

Full text available: pdf(370.62 KB) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

16 Improving software quality

Sharon Wheeler, Sheryl Duggins



April 1998 **Proceedings of the 36th annual Southeast regional conference ACM-SE 36**

Publisher: ACM Press

Full text available: [pdf\(1.43 MB\)](#) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

17 Self-adaptive multithreaded applications: a case for dynamic aspect weaving

Andreas Rasche, Wolfgang Schult, Andreas Polze

November 2005 **Proceedings of the 4th workshop on Reflective and adaptive middleware systems ARM '05**

Publisher: ACM Press

Full text available: [pdf\(442.90 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Shorter product cycles, the requirement for immediate reaction to cyber-attacks and the need for the adaptation to changing environmental conditions demand software reconfigurations to be performed at runtime, in order to reduce downtime. Especially long running applications, which have to provide continuous service should not be restarted for maintenance. They must be updated dynamically. We have developed a reconfiguration strategy allowing to identify valid reconfiguration points even in multi ...

Keywords: aspect oriented programming, dynamic reconfiguration, runtime software update

18 Software management

Cindy Dooling, Jeff White, Cathy Lee

November 2006 **Proceedings of the 34th annual ACM SIGUCCS conference on User services SIGUCCS '06**

Publisher: ACM Press

Full text available: [pdf\(66.43 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Software management problems? In this session presenters will discuss and demonstrate how Pima Community College (PCC) reigned in control of the software on 5,550 desktop and portable computers located over six campuses, two district offices, and nine center locations. To combat the lack of software standards and perceived legality issues, PCC developed a software plan with input from faculty, staff, and administrators. Prior to implementation of this plan, all users could purchase, download, or ...

Keywords: administrative support, audit, communication, computers, licenses, principles, software management, training

19 Deploying, updating, and managing tools for collecting software metrics

Alberto Sillitti, Barbara Russo, Paolo Zuliani, Giancarlo Succi

November 2004 **Proceedings of the 2004 workshop on Quantitative techniques for software agile process QUTE-SWAP '04**

Publisher: ACM Press

Full text available: [pdf\(351.02 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Collecting software engineering data is difficult due to the number of problems that researchers face in this activity. One of the most relevant is the ability to install and keep up-to-date the measurement tools installed in the production machines in order to collect such data. Even when a few machines are involved, maintaining all the tools required to collect data from the different development tools requires a full-time system administrator. Moreover, since most of these tools are research ...

Keywords: process monitoring, software metrics


20

Application of modern software techniques to modeling and simulation

Ronald M. Huhn, Edward R. Comer

January 1981 **Proceedings of the 13th conference on Winter simulation - Volume 1**
WSC '81

Publisher: IEEE Press

Full text available:  [pdf\(966.15 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

It is commonly agreed that software developments tend to be high risk activities; simulation is recognized as being even more "exciting". Great emphasis is being placed to develop methodologies which lower the risk of software development. Since a major portion of simulation activity is software oriented, it is natural to look to these modern software methodologies for solutions applicable to the modeling and simulation community. The total solution to current dilemmas is many y ...

Results 1 - 20 of 200

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2007 ACM, Inc.

[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Contact Us](#)

Useful downloads:  [Adobe Acrobat](#)  [QuickTime](#)  [Windows Media Player](#)  [Real Player](#)